SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis, Master's Thesis, … in Informatics

Evaluating learning algorithms: An efficient way/Efficient ways to find Regular Inductive Statements

Author

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis, Master's Thesis, ... in Informatics

Evaluating learning algorithms: An efficient way/Efficient ways to find Regular Inductive Statements

Titel der Abschlussarbeit

Author:	Author
Supervisor:	Supervisor
Advisor:	Advisor
Submission Date:	Submission date

I confirm that this bachelor's thesis, master's thesis, … is my own work and I have documented all sources and material used.

Munich, Submission date Author

Abstract

Acknowledgments

Contents

1 Introduction

As software systems grow in size and permeate more and more areas of our lives. Individuals and organizations use the majority of software in their systems. Thus the reliability and stability of the software testing are of major importance. Simulation and testing can detect bugs but not prove their absence. Such reactive systems, when no function is being computed, termination is usually undesirable. For this reason, we are interested in *property checking* or *model checking*. It has found a wide range of applications spanning from adaptive model checking.

Model checking is a powerful technique for automatic verification of finite state concurrent systems [CES09]. In this thesis, we only focus on *regular model checking*, which is a important framework for infinite state model-checking. The model is typically a regular transition system. It describes the potential behavior of discrete systems by representing it in finite state automaton. In [CES09] the author introduce an approach, that using the inductive statements for the regular transition for checking safety conditions. "Statement ψ is inductive if the transition relation only relates a state v satisfying ψ with states that also satisfy ψ. Thus, the set of all states that satisfy ψ over-approximates the set of all states reachable from v".

Based on automata learning, one can learn a set of inductive statements that are powerful enough to establish a given safety property. The learned language of inductive statements is a certificate of the correctness of the property. The purpose of this thesis is to collect and analyze empirical data on the performance of the learning algorithms such as L*, NL*, Kearns-Vazirani, Rivest-Schapire.

Structure of the thesis

In the first part of our thesis, we consider the regular transition system and the set of all inductive statements. In the second part of the thesis,

2 Preliminaries

In this section, we introduce some basic notions that we use throughout this thesis.

Finite automata

We use standard notions of finite automata. We distinguish between deterministic and non-deterministic automata to recognize regular languages of finite words.

Definition 2.1: *Deterministic finite automaton (DFA).*

A DFA is a quintuple $\mathcal{M} = (Q, q_0, \delta, \Sigma, F)$ *where* Q *is a finite set of states with a initial state* $q_0 \in Q$*. A set of input symbols called the alphabet* Σ*. A transition* $\delta : Q \times \Sigma \rightarrow Q$ *and a set of final states* F*. Let* $w = a_1 a_2 ... a_n$ *be a string over the alphabet* Σ*. The automaton* $\mathcal{M}$ *accepts* w *if a sequence of states,* $r_0, r_1, ... r_n$ *exist in* Q*:*

- $r_0 = q_0$
- $r_{i+1} = \delta(r_i, a_{i+1}), for\ i = 0, ..., n-1$
- $r_n \in F$

Definition 2.2: *Nondeterministic finite automaton (NFA).*

A NFA is a quintuple $\mathcal{N} = (Q, q_0, \Delta, \Sigma, F)$ *where* Q*,* Σ *and* F *are as for a DFA. Let* $w = a_1 a_2 ... a_n$ *be a string over the alphabet* Σ*. The automaton* $\mathcal{N}$ *accepts* w *if a sequence of states,* $r_0, r_1, ... r_n$ *exist in* Q*:*

- $r_0 = q_0$
- $r_{i+1} \in \Delta(r_i, a_{i+1}), for\ i = 0, ..., n-1$
- $r_n \in F$

3 Inductive statements for regular transition system

Related Work

Inductive statements for regular transition system was introduced by M.Sc. Welzel-Mohr.

4 Algorithmic Learning of Finite Automata

Learning automata is a computational model for solving problems, where an agent learns to optimize its behavior by interacting with an unknown environment. The agent, also known as a learner, observes the feedback from the teacher, updates its internal state, and adjusts its actions accordingly. The teacher This interaction process between the learner and the teacher is the primary mechanism of learning automata. In the field of automata learning, there are generally two distinct settings: active and passive learning. However, in this thesis, we only focus on the active learning. We do not introduce passive learning here but refer the interested reader to [CES09].

This chapter aims to provide a deeper understanding of the process of learning automata, including the roles and responsibilities of the teacher and learner.

4.1 The Teacher and Learner

In this learning scenario, the teacher is proficient in the language being taught and is responsible for answering any questions posed by the learner. The learner is given the opportunity to ask two types of queries - membership and equivalence. Membership queries are used to classify a word based on whether it belongs to the language being taught or not. Equivalence queries, on the other hand, are used to determine whether an assumed automaton is equivalent to the language the teacher has in mind. The learning process continues until the teacher answers an equivalence query positively.

Membership oracle The learner provides a word $w \in \Sigma^*$, the teacher replies "yes" or "no" depending on whether $w \in \mathcal{L}$ or not.

Equivalent oracle The learner conjectures a regular language, typically given as a DFA $\mathcal{M}$, and the teacher checks whether $\mathcal{M}$ is an equivalent description of the target language $\mathcal{L}$, otherwise return an counterexample $u \in \Sigma^*$ with $u \in \mathcal{L}(\mathcal{M}) \iff u \notin (\mathcal{L})$.

4.2 Algorithms

A learning algorithm—often called learner—learns a regular target language $\mathcal{L} \subset \Sigma^*$ over an a priori fixed alphabet Σ by actively querying a teacher. We apply several of these algorithms in the course of this thesis.

4.2.1 L*

4.2.2 NL*

4.2.3 Kearns-Vazirani

4.2.4 Rivest-Schapire

5 Implementation

We apply automata learning algorithms to solve the regular model checking problems as well as finding an inductive statements for regular transition system.

Membership query On a membership oracle, the learner provides a statement and asks the teacher if this statement whether inductive or not. As we described in Chapter 3, a statement I is *inductive* if, for any transition $v \rightsquigarrow u$ where u satisfies I, u also satisfies the statement. REF TO ABOVE DEFINITION. Therefore, one can simply implement the Membership Oracle by checking the acceptance of $\mathcal{M}$ while $\mathcal{M}$ is an automaton for $\overline{Inductive_{\mathcal{V}}(\mathcal{M})}$ and negating the answer.

6 Conclusion

We studied Regular Model Checking of safety properties. We evaluated the performance of our algorithms based on a prototype implementation.

Open Questions and Future Research

Abbreviations

List of Figures

List of Tables

Bibliography

[CES09] E. M. Clarke, E. A. Emerson, and J. Sifakis. “Model checking: algorithmic verification and debugging.” In: *Communications of the ACM* 52.11 (2009), pp. 74–84.